

Simulador de trajetória parabólica

Objetivo

Gerar uma **curva parabólica simulando a trajetória de um projétil**, com o aluno controlando o **ângulo** via potenciômetro, curva esta que será desenhada no **Serial Plotter**, simulando a **altura ao longo do tempo** ou da distância e dois LEDs comandados por sinais PWM para ilustrar na prática o ocorrido.

Montagem do circuito

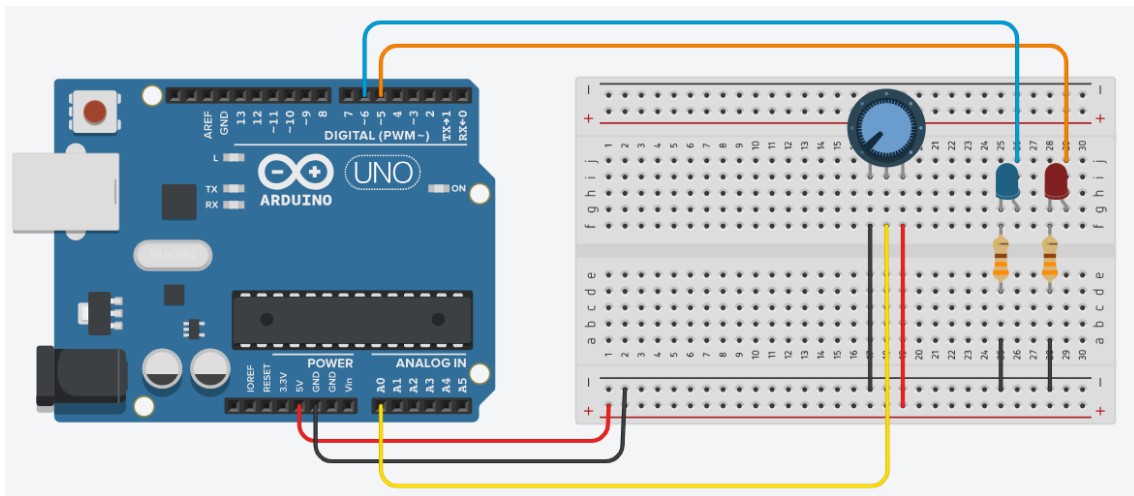


Figura 1 - Esquema e montagem do circuito.

Lista de materiais

1	Potenciômetro de 5k Ohms ou 10k Ohms
1	Arduino UNO
2	Resistores 330R (laranja, laranja, marrom)
1	LED vermelho 5mm ou 3mm
1	LED azul 5mm ou 3mm
1	Protoboard
	Cabinhos e fios para ligações

Teste e uso

Inicialmente gire o potenciômetro totalmente para o lado em que está o terminal GND. Ao ligar o circuito no computador, via cabo USB, abra o Plotter Serial indo em Ferramentas – Serial Plotter, com isso verá a formação de um gráfico na tela no formato

de uma curva que parte de um ponto a outro dentro de um intervalo de tempo. Gire o potenciômetro com calma e pare em certo ponto e veja a formação da curva, vai girando e observando o formato para cada ângulo de posicionamento do potenciômetro.

Na simulação de lançamento a velocidade do voo é calculada pela fórmula:

$$t_{total} = \frac{2v_0 \cdot \sin(\theta)}{g}$$

Onde:

V_0 = Velocidade de lançamento (que em nosso caso é fixa em 20 m/s);

θ = Ângulo de rotação do potenciômetro (ângulo inicial de lançamento do projétil);

g = Aceleração da gravidade terrestre (9,8 m/s²);

t = Tempo de duração da trajetória.

Para cada instante de tempo calculamos os valores de x e y com as seguintes fórmulas:

$$x = v_0 \cdot \cos(\theta) \cdot t$$

$$y = v_0 \cdot \sin(\theta) \cdot t - \frac{1}{2}gt^2$$

Código de programação

SimuladorTrajetoriaParabolica.ino

```
1  /*****
2  * Projeto: Estudo da trajetória parabólica em função do tempo *
3  * Desenvolvedor: Vanderlei Alves *
4  * Data: 27/06/2025 *
5  *****/
6  //Mapeamento dos pinos
7  #define POT_PIN A0
8  #define PIN_PWM_Y 5
9  #define PIN_PWM_X 6
10
11 //Constante da gravidade da terra
12 const float g = 9.8;
13
14 //Variáveis globais
15 int pwm_y,
16     |   pwm_x;
17
18 void setup()
19 {
20     //Inicialização da comunicação serial
21     Serial.begin(9600);
22
23     //Aguarda 1 segundo para estabilizar as tensões
24     delay(1000);
25 }
26
27 void loop()
28 {
29     //Leitura do potenciômetro
30     int leitura = analogRead(POT_PIN);
31
32     //Conversão dos valores para ângulos
33     float angulo = map(leitura, 0, 1023, 10, 80); //Ângulo de 10° a 80°
34
35     //Conversão dos ângulos em radianos
36     float rad = angulo * 3.14159 / 180.0;
37
38     float v0 = 20.0; //Velocidade fixa de 20 m/s do objeto lançado
39     float t_total = (2 * v0 * sin(rad)) / g; //Tempo total de voo
40     float passo_tempo = t_total / 50.0; //50 pontos
41
42     Serial.println("\x\tty"); //Cabeçalho para o Serial Plotter
43 }
```

```

44 for (float t = 0; t <= t_total; t += passo_tempo)
45 {
46     //Cálculo dos valores de x e y em função do tempo
47     float x = v0 * cos(rad) * t;
48     float y = v0 * sin(rad) * t - 0.5 * g * t * t;
49     if (y < 0) y = 0; //Evita valores negativos
50
51     //Representação de x e y com os LEDs
52     pwm_x = map(x, 0, 14, 0, 255);
53     analogWrite(PIN_PWM_X, pwm_x);
54
55     pwm_y = map(y, 0, 20, 0, 255);
56     analogWrite(PIN_PWM_Y, pwm_y);
57
58     //Apresentação do gráfico no Plotter Serial
59     Serial.print(x);
60     Serial.print("\t");
61     Serial.println(y);
62
63     delay(50); //Tempo entre pontos no plotter
64 }
65
66 delay(2000); //Espera para novo gráfico
67 }

```